Improving the Performance of Mesh Optimization Techniques

Paul Hovland and Todd Munson*, Argonne National Laboratory

Summary

Optimizing the element quality in a mesh can significantly reduce the time required to find an approximate solution to a partial differential equation. Calculating the improved mesh in the least amount of time requires both an efficient algorithm and an efficient implementation. The code Argonne researchers have developed to compute an optimal mesh has successfully reduced the execution time of a spectral-element method applied to a fluid dynamics application by 30 percent.

Discretization methods are common techniques for computing approximate solutions to partial differential equations. These methods decompose the given domain into a finite set of elements, triangles, or tetrahedrons, for example, to produce a mesh used within the approximation scheme. Several factors affect the accuracy of the solution to the partial differential equation computed: the degree of the approximation scheme and the number of elements in the mesh, and the quality of the mesh. Optimizing the quality of the mesh prior to computing the approximate solution can improve the condition number of the linear systems solved, reduce the time taken to compute the solution, and increase the numerical accuracy.

The savings in computational time from using the optimized mesh can be substantial. One application we investigated with Paul Fischer (ANL) and S. Balachandar and Lin Zhang (UIUC) was to solve the Navier-Stokes equations for a fluid containing a dilute suspension of particles. The approximate solution was obtained by applying a spectral-element method to a hexahedral mesh. The top of Figure 1 depicts their original mesh, while the bottom shows the mesh after shape-quality improvement techniques have been applied. The original mesh has many regular elements, while the optimized mesh loses much of this structure. However, their spectral-element method required 29 hours to compute a solution when using the original mesh, but only 20 hours when using the optimized mesh, a 30 percent reduction in time.

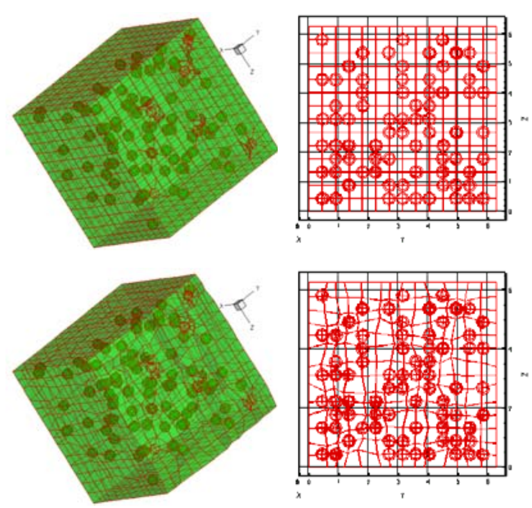

Figure 1. Original mesh and side view (top), and optimized mesh and side view (bottom) for a fluid dynamics application with a dilute suspension of particles.

* Mathematics and Computer Science Division, (630) 252-4279, tmunson@mcs.anl.gov

The optimization problem we solve computes positions for the vertices in the mesh to improve the average element quality by using the inverse mean-ratio metric, a shape-quality metric measuring the distance between a trial element and an ideal element, a regular tetrahedron, for example. The objective function for the resulting optimization problem is nonconvex and consists of the sum of many fractional terms. Included in the optimization problem are periodic boundary conditions and constraints restricting the vertices to the planes and spheres defining the domain.

The computational properties of an inexact Newton method with a line search have been extensively studied when solving a simplified optimization problem where the positions of the vertices on the boundary of the mesh are fixed. The conjugate gradient method with a block Jacobi preconditioner is applied to solve the systems of equations. To improve the performance of this code, we have applied several techniques: using a block sparse matrix data structure to store the Hessian, reordering the problem data, and preconditioning the iterative method. These techniques can significantly reduce the computational time, especially for large problem instances.

Modern microprocessors are highly sensitive to the spatial and temporal locality of data sets. Therefore, reordering the vertices and elements in a mesh can have a significant impact on performance. We have developed metrics and models for mesh reordering and have investigated the performance of several reordering algorithms. Modeling the mesh as a hypergraph is critical and can lead to performance improvements of nearly 50 percent. Figures 2 and 3 show the effects of the data-reordering algorithms on the sparsity pattern and execution time. The performance of the gradient and Hessian evaluations was improved by approximately 20 percent using the reverse mode of automatic differentiation. Another 10 percent gain in overall performance was achieved with an improved implementation of the reciprocal cube root function.

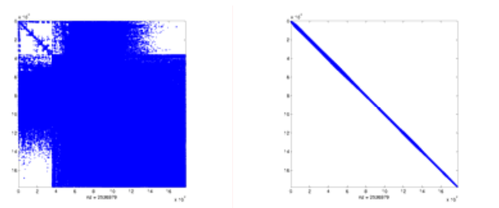

Figure 2. Sparsity pattern of the Hessian matrix for original (left) and reordered problems (right) for the duct8 mesh.

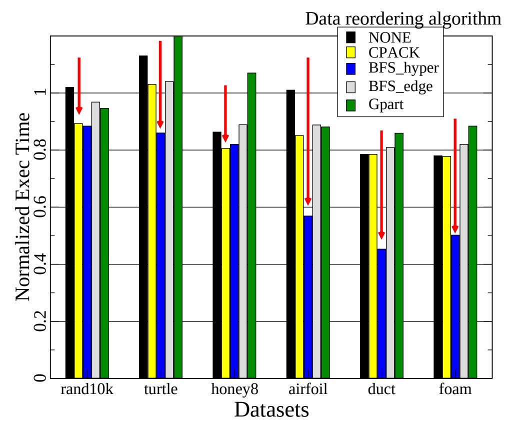

Figure 3. Effects of data-reordering algorithms on optimization time.

We plan to extend our work to constrained optimization problems where certain vertices are required to lie on the boundary of the domain. This work will require either access to the geometry or a mechanism to infer geometric features from the mesh data. These constrained problems are more difficult to solve. We will investigate techniques to improve performance of the resulting code.

For further information on this subject contact:
Todd Munson
Argonne National Laboratory
Mathematics and Computer Science Division
tmunson@mcs.anl.gov
(630) 252-4279